ch04output

2024-02-21

```
# Code from Chapter 4 of R Companion for Sampling: Design and Analysis by
# Yan Lu and Sharon L. Lohr
# All code is presented for educational purposes only and without warranty.

##### Install the R packages needed for the chapter

library(survey)
```

```
## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart
```

```
library(sampling)
```

```
##
## Attaching package: 'sampling'

## The following objects are masked from 'package:survival':
##
##     cluster, strata
```

```
library(SDAResources)

########## Ratio Estimation ##########

##### Examples 4.2 and 4.3

data(agsrs)
n<-nrow(agsrs) #300
agsrs$sampwt <- rep(3078/n,n)
agdsrs <- svydesign(id = ~1, weights=~sampwt, fpc=rep(3078,300), data = agsrs)
agdsrs
```

```
## Independent Sampling design
## svydesign(id = ~1, weights = ~sampwt, fpc = rep(3078, 300), data = agsrs)
```

```
# correlation of acres87 and acres92
cor(agsrs$acres87,agsrs$acres92)
```

```
## [1] 0.995806
```

```
# estimate the ratio acres92/acres87
sratio<-svyratio(numerator = ~acres92, denominator = ~acres87,design = agdsrs)
sratio
```

```
## Ratio estimator: svyratio.survey.design2(numerator = ~acres92, denominator = ~acres87,
##     design = agdsrs)
## Ratios=
##           acres87
## acres92 0.9865652
## SEs=
##             acres87
## acres92 0.005750473
```

```
confint(sratio, df=degf(agdsrs))
```

```
##                     2.5 %    97.5 %
## acres92/acres87 0.9752487 0.9978818
```

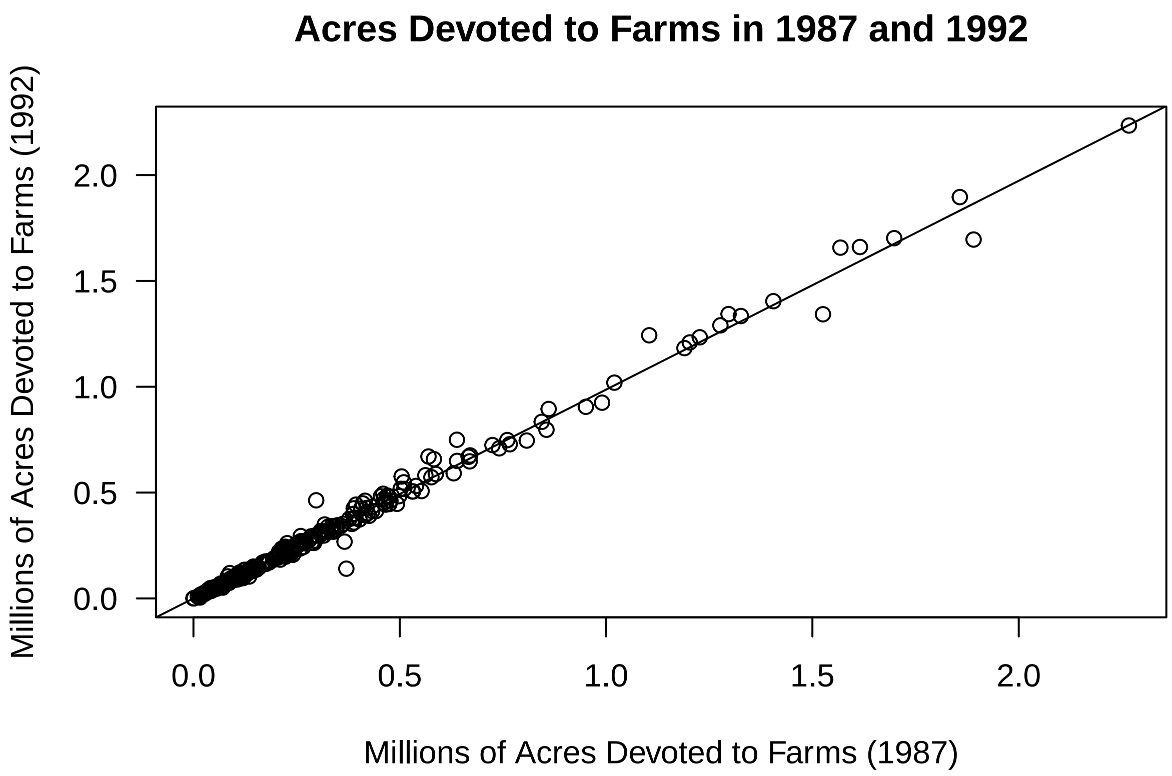
```
# provide the population total of x
xpoptotal <- 964470625
# Ratio estimate of population total
predict(sratio,total=xpoptotal)
```

```
## $total
##           acres87
## acres92 951513191
##
## $se
##         acres87
## acres92 5546162
```

```
# Ratio estimate of population mean
predict(sratio,total=xpoptotal/3078)
```

```
## $total
##          acres87
## acres92 309133.6
##
## $se
##          acres87
## acres92 1801.872
```

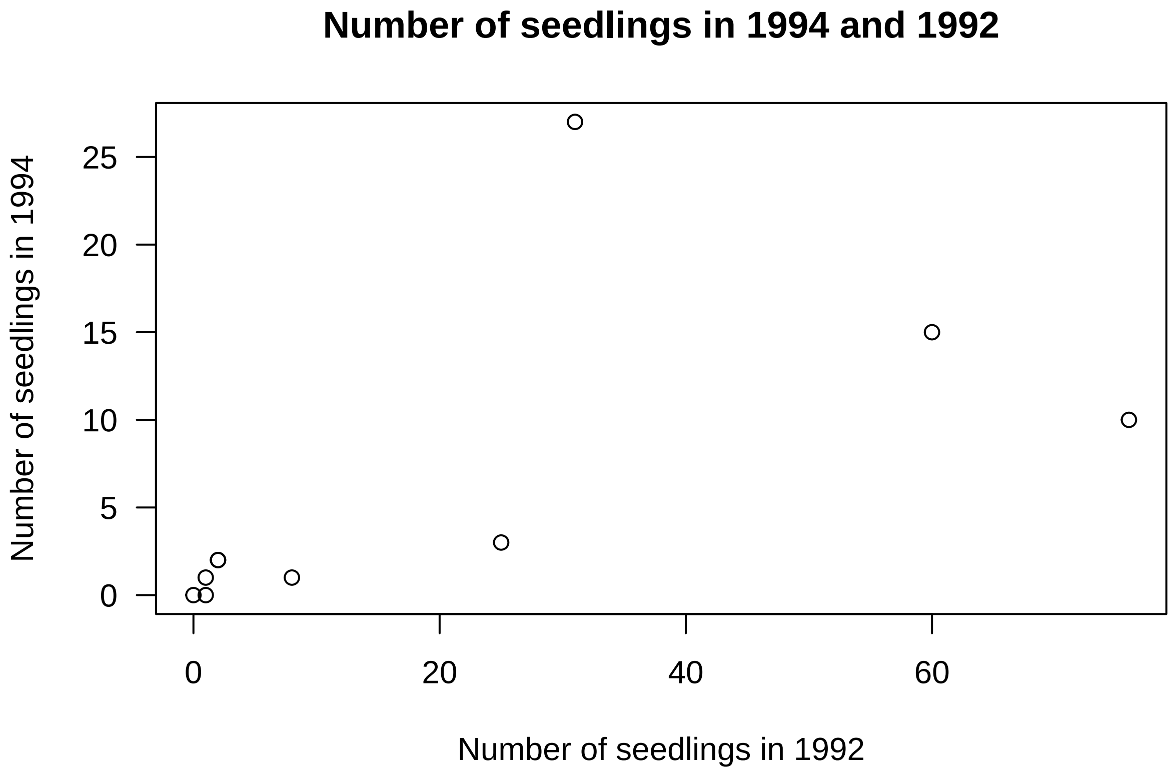
```
# draw the scatterplot
par(las=1) # make tick mark labels horizontal (optional)
plot(x=agsrs$acres87/1e6,y=agsrs$acres92/1e6,
  xlab="Millions of Acres Devoted to Farms (1987)",
  ylab = "Millions of Acres Devoted to Farms (1992)",
  main = "Acres Devoted to Farms in 1987 and 1992")
# draw line through origin with slope Bhat
abline(0,coef(sratio))
```



```
##### Example 4.5

# scatterplot and correlation of seed92 and seed94
data(santacruz)
santacruz
```

```
##    tree seed92 seed94
## 1     1      1      0
## 2     2      0      0
## 3     3      8      1
## 4     4      2      2
## 5     5     76     10
## 6     6     60     15
## 7     7     25      3
## 8     8      2      2
## 9     9      1      1
## 10    10      31      27
```

```
plot(santacruz$seed92,santacruz$seed94,
     main="Number of seedlings in 1994 and 1992",
     xlab="Number of seedlings in 1992",ylab="Number of seedlings in 1994")
```



```
cor(santacruz$seed92,santacruz$seed94)
```

```
## [1] 0.6106537
```

```
nrow(santacruz) #10
```

```
## [1] 10
```

```
santacruz$sampwt <- rep(1,nrow(santacruz))
design0405 <- svydesign(ids = ~1, weights = ~sampwt, data = santacruz)
design0405
```

```
## Independent Sampling design (with replacement)
## svydesign(ids = ~1, weights = ~sampwt, data = santacruz)
```

```
#Ratio estimation using number of seedlings of 1992 as auxiliary variable
sratio3<-svyratio(~seed94, ~seed92,design = design0405)
sratio3
```

```
## Ratio estimator: svyratio.survey.design2(~seed94, ~seed92, design = design0405)
## Ratios=
##           seed92
## seed94 0.2961165
## SEs=
##           seed92
## seed94 0.1152622
```

```
confint(sratio3, df=10-1)
```

```
##                      2.5 %    97.5 %
## seed94/seed92 0.03537532 0.5568577
```

```
########## Regression Estimation ##########

##### Example 4.7

data(deadtrees)
head(deadtrees)
```

```
##   photo field
## 1    10    15
## 2    12    14
## 3     7     9
## 4    13    14
## 5    13     8
## 6     6     5
```

```
nrow(deadtrees) # 25
```

```
## [1] 25
```

```
# Fit with survey regression
dtree<- svydesign(id = ~1, weight=rep(4,25), fpc=rep(100,25), data = deadtrees)
dtree
```

```
## Independent Sampling design
## svydesign(id = ~1, weight = rep(4, 25), fpc = rep(100, 25), data = deadtrees)
```

```
myfit1 <- svyglm(field~photo, design=dtree)
summary(myfit1) # displays regression coefficients
```

```
##
## Call:
## svyglm(formula = field ~ photo, design = dtree)
##
```

```
## Survey design:
## svydesign(id = ~1, weight = rep(4, 25), fpc = rep(100, 25), data = deadtrees)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0593     1.3930   3.632   0.0014 **
## photo         0.6133     0.1259   4.870 6.44e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for gaussian family taken to be 5.548341)
##
## Number of Fisher Scoring iterations: 2
```

```
confint(myfit1,df=23) # df = 25-2
```

```
##                  2.5 %   97.5 %
## (Intercept) 2.1777362 7.940848
## photo       0.3527717 0.873777
```

```
# Regression estimate of population mean field trees
newdata <- data.frame(photo=11.3)
predict(myfit1, newdata)
```

```
##     link    SE
## 1 11.989 0.418
```

```
confint(predict(myfit1, newdata),df=23)
```

```
##      2.5 %   97.5 %
## 1 11.12455 12.85404
```

```
# Estimate total field tree, add population size in total= argument
newdata2 <- data.frame(photo=1130)
predict(myfit1, newdata2, total=100)
```

```
##     link     SE
## 1 1198.9 41.802
```

```
confint(predict(myfit1, newdata2,total=100),df=23)
```

```
##      2.5 %   97.5 %
## 1 1112.455 1285.404
```

```
########## Domain Estimation ##########

##### Example 4.8

agsrsnew<-agsrs #copy agsrs as agsrsnew, since we want to create a new column
# we calculated sampwt in the first code in this chapter
# define new variable farmcat
agsrsnew$farmcat<-rep("large",n)
agsrsnew$farmcat[agsrsnew$farms92 < 600] <- "small"
head(agsrsnew)
```

```
##              county state acres92 acres87 acres82 farms92 farms87 farms82
## 1     COFFEE COUNTY    AL  175209  179311  194509     760     842     944
## 2    COLBERT COUNTY    AL  138135  145104  161360     488     563     686
## 3      LAMAR COUNTY    AL   56102   59861   72334     299     362     447
## 4    MARENGO COUNTY    AL  199117  220526  231207     434     471     622
## 5     MARION COUNTY    AL   89228  105586  113618     566     658     748
## 6 TUSCALOOSA COUNTY    AL   96194  120542  134616     436     521     650
##   largef92 largef87 largef82 smallf92 smallf87 smallf82 region sampwt farmcat
## 1       29       28       21       57       47       66      S  10.26   large
## 2       37       41       42       12       44       47      S  10.26   small
## 3        4        4        3       16       20       30      S  10.26   small
## 4       48       66       62       14       11       28      S  10.26   small
## 5        7        9        9       11       23       27      S  10.26   small
## 6       20       17       23       18       32       29      S  10.26   small
```

```
dsrsnew <- svydesign(id = ~1, weights=~sampwt, fpc=rep(3078,300), data=agsrsnew)
dsrsnew
```

```
## Independent Sampling design
## svydesign(id = ~1, weights = ~sampwt, fpc = rep(3078, 300), data = agsrsnew)
```

```
# domain estimation for large farmcat with subset statement
dsub1<-subset(dsrsnew,farmcat=='large')   # design info for domain large farmcat
smean1<-svymean(~acres92,design=dsub1)
smean1
```

```
##           mean    SE
## acres92 316566 21553
```

```
df1<-sum(agsrsnew$farmcat=='large')-1 #calculate domain df if desired
df1
```

```
## [1] 128
```

```
confint(smean1, level=.95,df=df1) # CI
```

```
##            2.5 %   97.5 %
## acres92 273918.9 359212.4
```

```
stotal1<-svytotal(~acres92,design=dsub1)
stotal1
```

```
##             total       SE
## acres92 418987302 38938277
```

```r
confint(stotal1, level=.95,df=df1)
```

```
##              2.5 %    97.5 %
## acres92 341941269 496033335
```

```r
# domain estimation for small farmcat
dsub2<-subset(dsrsnew,farmcat=='small')  # design info for domain small farmcat
smean2<-svymean(~acres92,design=dsub2)
smean2
```

```
##           mean    SE
## acres92 283814 28852
```

```r
df2<-sum(agsrsnew$farmcat=='small')-1 #calculate domain df if desired
confint(smean2, level=.95,df=df2) #CI
```

```
##             2.5 %   97.5 %
## acres92 226858.9 340768.5
```

```r
stotal2<-svytotal(~acres92,design=dsub2)
stotal2
```

```
##              total       SE
## acres92 497939808 55919525
```

```r
confint(stotal2, level=.95,df=df2)
```

```
##              2.5 %    97.5 %
## acres92 387553732 608325884
```

```r
# use svyby function
bothtot<-svyby(~acres92,by=~factor(farmcat),design=dsrsnew,svytotal)
bothtot
```

```
##       factor(farmcat)   acres92       se
## large           large 418987302 38938277
## small           small 497939808 55919525
```

```r
confint(bothtot,level=.95)
```

```
##            2.5 %    97.5 %
## large 342669682 495304922
## small 388339553 607540062
```

```r
bothmeans<-svyby(~acres92,by=~factor(farmcat),design=dsrsnew,svymean)
bothmeans
```

```
##       factor(farmcat)  acres92       se
## large           large 316565.7 21553.21
## small           small 283813.7 28852.24
```

```
confint(bothmeans,level=.95)
```

```
##          2.5 %   97.5 %
## large 274322.1 358809.2
## small 227264.4 340363.1
```

```
########## Poststratification ##########

##### Example 4.9

data(agsrs)
dsrs <- svydesign(id = ~1, weights=rep(3078/300,300), fpc=rep(3078,300),
                  data = agsrs)
# Create a data frame that gives the population totals for the poststrata
pop.region <- data.frame(region=c("NC","NE","S","W"), Freq=c(1054,220,1382,422))
# create design information with poststratification
dsrsp<-postStratify(dsrs, ~region, pop.region)
summary(dsrsp)
```

```
## Independent Sampling design
## postStratify(dsrs, ~region, pop.region)
## Probabilities:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.09242 0.09407 0.09407 0.09771 0.10152 0.10909
## Population size (PSUs): 3078
## Data variables:
##  [1] "county"   "state"    "acres92"  "acres87"  "acres82"  "farms92"
##  [7] "farms87"  "farms82"  "largef92" "largef87" "largef82" "smallf92"
## [13] "smallf87" "smallf82" "region"
```

```
1/unique(dsrsp$prob)  # See the poststratified weight for each region
```

```
## [1] 10.630769 10.820513  9.850467  9.166667
```

```
svymean(~acres92, dsrsp)
```

```
##           mean    SE
## acres92 299778 17513
```

```
svytotal(~acres92, dsrsp)
```

```
##             total       SE
## acres92 922717031 53906392
```

```
########## Ratio Estimation with Stratified Sampling ##########

##### Combined ratio estimator

data(agstrat)
```

```r
popsize_recode <- c('NC' = 1054, 'NE' = 220, 'S' = 1382, 'W' = 422)
agstrat$popsize <- popsize_recode[agstrat$region]
# input design information for agstrat
dstr <- svydesign(id = ~1, strata = ~region, fpc = ~popsize, weight = ~strwt,
                  data = agstrat)
# now compute the combined estimator of the ratio
combined<-svyratio(~ acres92,~acres87,design = dstr)
combined
```

```
## Ratio estimator: svyratio.survey.design2(~acres92, ~acres87, design = dstr)
## Ratios=
##            acres87
## acres92 0.9899971
## SEs=
##              acres87
## acres92 0.006187757
```

```r
# we can get the combined ratio estimator of the population total
# with the predict function
predict(combined,total=964470625)
```

```
## $total
##            acres87
## acres92 954823130
##
## $se
##          acres87
## acres92 5967910
```

```r
##### Separate ratio estimator

separate<-svyratio(~acres92,~acres87,design = dstr,separate=TRUE)
separate
```

```
## Stratified ratio estimate: svyratio.survey.design2(~acres92, ~acres87, design = dstr, separate = TRU
## Ratio estimator: Stratum == "NC"
## Ratios=
##            acres87
## acres92 0.9750666
## SEs=
##              acres87
## acres92 0.005483458
## Ratio estimator: Stratum == "NE"
## Ratios=
##            acres87
## acres92 0.8956073
## SEs=
##              acres87
## acres92 0.008853011
## Ratio estimator: Stratum == "S"
## Ratios=
##            acres87
## acres92 0.9935483
## SEs=
##             acres87
## acres92 0.01418835
## Ratio estimator: Stratum == "W"
## Ratios=
##           acres87
## acres92 1.011974
## SEs=
##             acres87
## acres92 0.01169809
```

```
#  Define the stratum totals for acres87 as a list:
stratum.xtotals <- list(NC=350474227,NE=22033421,S=280631939,W=311331038)
predict(separate,stratum.xtotals)
```

```
## $total
##            acres87
## acres92 955349448
##
## $se
##          acres87
## acres92 5731438
```

```
########## Model-Based Ratio and Regression Estimation ##########

##### Example 4.11

data(agsrs)
# define weights to use for weighted least squares analysis
agsrs$recacr87<-agsrs$acres87
agsrs$recacr87[agsrs$acres87!=0] <- 1/agsrs$acres87[agsrs$acres87!=0]
agsrs$recacr87[agsrs$acres87==0] <- NA
# fit weighted least squares model without intercept
fit<-lm(acres92~acres87-1,weights=recacr87,data=agsrs)
summary(fit)
```

```
##
## Call:
## lm(formula = acres92 ~ acres87 - 1, data = agsrs, weights = recacr87)
##
## Weighted Residuals:
##    Min     1Q Median     3Q    Max
## -369.9  -22.2   -5.8   10.8  311.7
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## acres87 0.986565   0.004844   203.7   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 46.1 on 298 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.9929, Adjusted R-squared:  0.9928
## F-statistic: 4.149e+04 on 1 and 298 DF,  p-value: < 2.2e-16
```

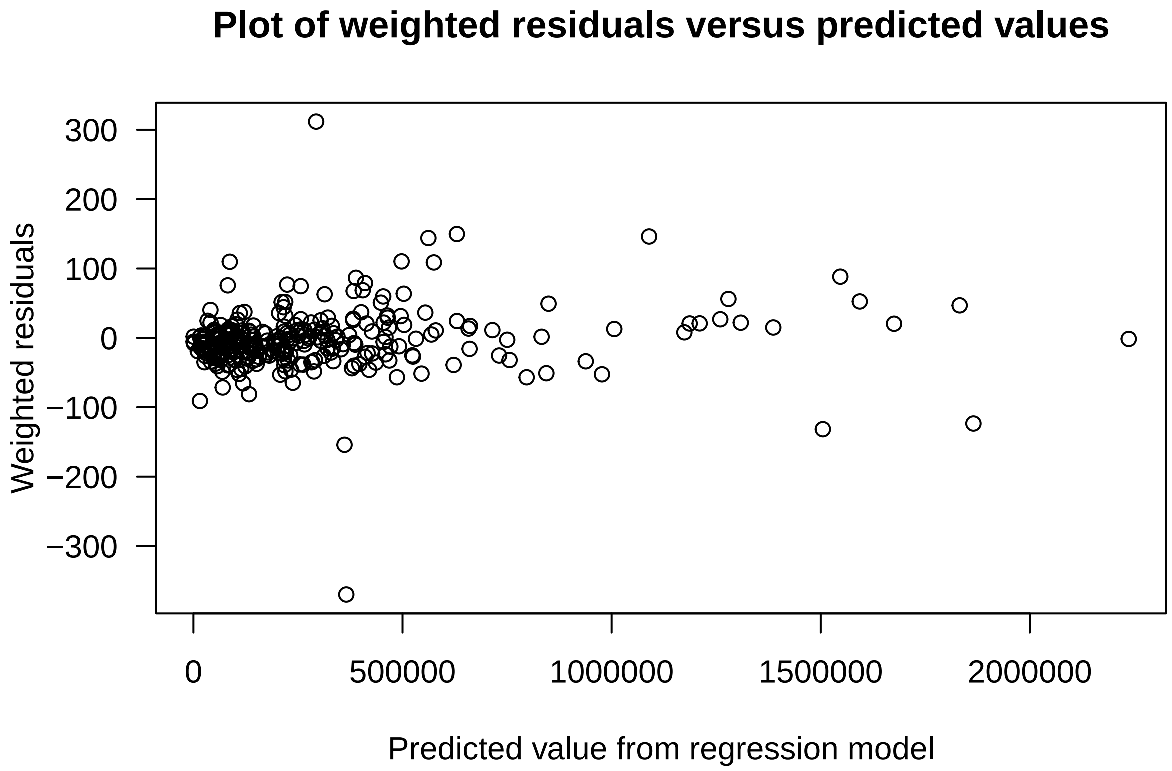
```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: acres92
##            Df   Sum Sq  Mean Sq F value    Pr(>F)
## acres87     1 88168461 88168461   41487 < 2.2e-16 ***
## Residuals 298   633307     2125
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
# find predicted value at population total for x
newdata3 <- data.frame(acres87=964470625)
predict(fit, newdata3, se.fit=TRUE)
```

```
## $fit
##         1
## 951513191
##
## $se.fit
## [1] 4671509
##
## $df
## [1] 298
##
## $residual.scale
## [1] 46.0998
```

```
# plot weighted residual versus predicted values
wresid<-fit$residuals*sqrt(fit$weights)
par(las=1)
plot(fit$fitted.values, wresid,
     main="Plot of weighted residuals versus predicted values",
     xlab="Predicted value from regression model",
     ylab="Weighted residuals")
```



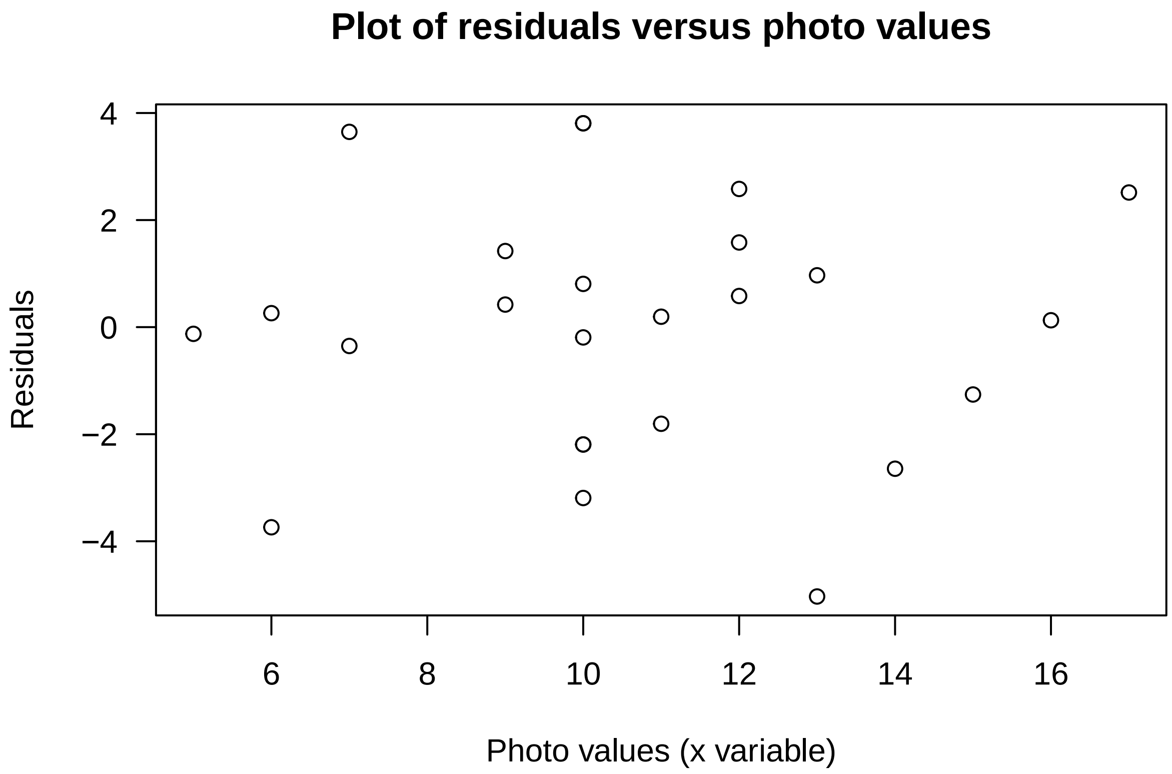
```
##### Example 4.12

data(deadtrees)
# Fit with lm
fit2 <- lm(field~photo, data=deadtrees)
summary(fit2)
```

```
##
## Call:
## lm(formula = field ~ photo, data = deadtrees)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0319 -1.8053  0.1947  1.4212  3.8080
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.0593     1.7635   2.869 0.008676 **
## photo         0.6133     0.1601   3.832 0.000854 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 23 degrees of freedom
## Multiple R-squared:  0.3896, Adjusted R-squared:  0.3631
## F-statistic: 14.68 on 1 and 23 DF,  p-value: 0.0008538
```

```
# Estimate mean field trees
newdata <- data.frame(photo=11.3)
predict(fit2, newdata,se.fit=TRUE)
```

```
## $fit
##        1
## 11.98929
##
## $se.fit
## [1] 0.4941007
##
## $df
## [1] 23
##
## $residual.scale
## [1] 2.406153
```

```
# plot residuals versus predicted values
plot(deadtrees$photo, fit2$residuals,
     main="Plot of residuals versus photo values",
     xlab="Photo values (x variable)",
     ylab="Residuals")
```


Plot of residuals versus photo values
Residuals
Photo values (x variable)